311 Mid Male jan 2023

Question 1 .... **5 points**

Prove or disprove that:

(a) [1 point] $n(n+1)/2 \in O(n^2)$ correct

$$\frac{n(n+1)}{2} = \frac{n^2+n}{2} = \frac{n^2}{2} + \frac{n}{2} \le n^2 \quad \forall\ n \ge 1$$

by using playnomial thereome ← $C = 1$ $n_0 = 1$

(b) [1 point] $n(n+1)/2 \in \Omega(n^2)$ by using polynomial therom $n_0 = \left|\frac{\sum_{i=0}^{k-1} a_i}{a_k - C}\right|$ $C = \frac{1}{4}$

$$= \frac{n^2}{2} + \frac{n}{2} \ge \frac{1}{4} n^2$$

$$n_0 = \frac{\frac{1}{2}}{\frac{1}{2} - \frac{1}{4}} = \frac{4}{2} = 2$$

$$\frac{n^2}{2} + \frac{n}{2} \ge \frac{1}{4} n^2 \quad \forall n \ge 2$$

$C = \frac{1}{4}$ $n_0 = 2$ correct

(c) [1 point] $n(n+1)/2 \in \Theta(n^2)$

frome previous Q a,b we can see that $C_1 = \frac{1}{4}$ $C_2 = 1$ $\max(n_0[1,2]) = 2$

$n_0 = 2$

correct

(d) [1 point] $n(n+1)/2 \in o(n^2)$

uncorrect

by defintion of small-o ~~the~~ All constants $c > 0$ should accepet $o(n^2)$ جميع الثوابت تقبل o

$$\frac{n^2}{2} + \frac{n}{2} < n^2 \quad \text{when } c = 1 \quad n = 1$$

$$1 \not< 1 \qquad \text{So } \frac{n(n+1)}{2} \notin o(n^2)$$

(c) [1 point] $n(n+1)/2 \in \omega(n)$

by defin~~it~~ton small $-\omega$ All constants should accept the $\omega(n)$ and this is true

$\frac{n^2}{2} + \frac{n}{2} > n \quad \forall n \geq 1 \; \forall c > 0$

correct

Question 2..4 points

For each of the following functions, indicate the class $\Theta(g(n))$ the function belongs to. Use the simplest $g(n)$ possible in your answers.

(a) [1 point] $f(n) = (n^2+1)^{10}$

$f(n) \in \Theta(n^{12})$ $\quad n^{20}$

-1

(b) [1 point] $f(n) = 2n\log(n+2)^2 + (n+2)^2\log(\frac{n}{2})$

$f(n) = 4n\log(n+2) + (n^2+4n+4)(\log n - \log 2)$

$f(n) = 4n\log(n+2) + n^2\log n + 4n\log n + 4\log n$

$f(n) \in \Theta(n^2\log n)$

(c) [1 point] $f(n) = \sqrt{10n^2+7n+3}$

$f(n) \in \Theta(n)$

(d) [1 point] $f(n) = \log^2(n).\log(n^2)$

$f(n) \in \Theta(\log^3(n))$

Question 3...2 points

Solve the following recurrence relation by using the backward substitution method.

$$\begin{cases} T(n) = 3T(n/5) + n & \text{for } n > 1 \\ T(1) = 0 \end{cases}$$

$T(n) = 3T(n/5) + n$

$T(n) = 3\left[3T\left(\frac{n}{5^2}\right) + \frac{n}{5}\right] + n$

$T(n) = 3^2 T\left(\frac{n}{5^2}\right) + \frac{3}{5}n + n$

$= 3^2\left[3T\left(\frac{n}{5^3}\right) + \frac{n}{5^2}\right] + \frac{3}{5}n + n$

$= 3^3 T\left(\frac{n}{5^3}\right) + \left(\frac{3}{5}\right)^2 n + \left(\frac{3}{5}\right)n + n$

$\vdots$

after k step $T(n) = 3^k T\left(\frac{n}{5^k}\right) + \sum_{i=0}^{k-1}\left(\frac{3}{5}\right)^i n$

$\frac{n}{5^k} = 1$

$= 3^k T\left(\frac{n}{5^k}\right) + n\left(\frac{1-\left(\frac{3}{5}\right)^k}{1-\frac{3}{5}}\right)$

$k = \log_5 n$

$T\left(\frac{n}{5^k}\right) = 0$

$= 3^k . T\left(\frac{n}{5^k}\right) + \frac{5}{2}n\left(1 - \left(\frac{3}{5}\right)^k\right)$

$\leq \frac{5}{2}n \in O(n)$

Question 4...4 points

Consider the following algorithm.

Algorithm 1 Unknown($A[0..n-1]$) ▷ Input: An array $A[0..n-1]$ of n real numbers

1: $minval \leftarrow A[0]$ $\longrightarrow 1$
2: $maxval \leftarrow A[0]$ $\longrightarrow 1$
3: **for** $i \leftarrow 1, n-1$ **do** $\longrightarrow n$
4:     **if** $A[i] < minval$ **then** $\longrightarrow n-1$
5:         $minval \leftarrow A[i]$ $\longrightarrow n-1$
6:     **end if**
7:     **if** $A[i] > maxval$ **then** $\longrightarrow n-1$
8:         $maxval \leftarrow A[i]$ $\longrightarrow n-1$
9:     **end if**
10: **end for**
11: **return** $maxval - minval$ $\longrightarrow 1$

$f(n) = 5n - 1$

maxmum ~~[illegible]~~

(a) [1 point] What does this algorithm compute?

this algorithm ditermaine the max value in the array and ninimum value of the array and return the deference between them

max val − min val

(b) [1 point] What is its basic operation?

the comparison in step 4 and step 7

المسوحة ضوئيا بـ CamScanner

(c) [2 points] Give the best-case and worst-case time complexities of this algorithm in asymptotic notation. here the best-case and the worst-case are the same

$f(n) = 5n - 1 \leq 5n \quad \forall n \geq n$

$c = 5 \quad n_0 = 1$

$(5n - 1) \in O(n)$ in worst case and best case

Question 5 .... *6 points*

Consider the following recursive algorithm for computing the sum of the first n squares of integers: $S(n) = 1^2 + 2^2 + ... + n^2$.

Algorithm 2 $S(n)$ ▷ Input: A positive integer n ▷ Output: the sum of the first n squares

```
1: if n = 1 then        → 1
2:     return 1         → 1      } ⇒ 0
3: else
4:     return S(n − 1) + n ∗ n    → T(n−1) + 2
5: end if                 1  1
```

(a) [½ point] What is the algorithm's basic operation?

the basic operation here is the summation and multiplication

Step 4

Artimetic

assume

S $T(1) = 0$

$n-k=1$

$k = n-1$

(b) [2 points] Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.

$T(n) = T(n-1) + 2$
$= [T(n-2) + 2] + 2$
$= [T(n-3) + 2] + 2 + 2$

after k step $\vdots$

$T(n) = T(n-k) + 2k$

$T(n) = 0 + 2(n-1) = 2n-2 \in O(n)$

(c) [2 points] Give a pseudocode of a non-recursive version of this algorithm.

```
procedure m(n)
S ← 0

for i=1..n do
    S ← S + i*i
end for
return S
```

$n+1$

$2n$

$0 + 1*1 + 2*2 + \dots + n*n$

(d) [½ point] What is the worst-case time complexity of the algorithm in (c)?

the worst time coplexity is $O(n)$.

(e) [1 point] What is the suitable algorithm for computing $S(n)$ (recursive or non-recursive)? Justify your answer.

bothe algorithms give us same time complixty but the recursive one ~~are~~ take more space complixty than non recursive

Question 6..4 points

(a) [3 points] Describe an algorithm that takes a list of n positive integers and finds the location of the last even integer in the list. It returns 0 if there are no even integers in the list.

we can seb the last ~~element~~ index in the list and put it in a variable and check the value at index n in ~~for~~ recursive or non recursive as for loop and the loop stop when ~~we~~ ~~find~~ the condition is true and return the index in the array

Linear search from last

(b) [1 point] Give the best-case and worst-case time complexities of your algorithm in asymptotic notation. Show all details.

the best-case is when the last index is even $\longrightarrow \in O(1)$ ×

$\boxed{\text{لا يقل عن } 1}$ $\Omega(1)$ عند الأخير أو قبل الأخير وهكذا

the worst-case is when the first index or there is no even in the list

$\longrightarrow \in O(n)$

$\boxed{\text{لا يزيد عن } n}$

-1/2

Q6-a

```
Alg lasteven(A[1..n])
{

  for I = n downTo 1
   {
    If(A[I] mod2 = 0)
      return I
   }
   return 0
}
```

Another solution

```
Alg lasTeven(A[1..n])
{
    I=n
    while(I > 0 and A[I] mod2 ≠ 0)
      I = I - 1

    If(I > 0)
      return I
    Else
      return 0
}
```

Question 4

Prove/disprove each of the following by finding appropriate values for c and n_0.

a) $n(n+1)/2 = O(n^2)$ // ignore negative
b) $2n^3 - 10n^2 + 2 = \Omega(n^3)$.

Question 5

For each of the following functions, indicate the class $\Theta(g(n))$ the function belongs to. (Use the simplest g(n) possible in your answers.) Prove your assertions.

a. $(n^2+1)^{10}$

حلها ورا الخلف b. $\sqrt{10n^2 + 7n + 3}$

(a) (i) $(n^2+1)^{10} \le (n^2+n^2)^{10}$

$\le (2n^2)^{10}$

$\le 2^{10} \cdot (n^2)^{10}$

$\le 2^{10} \cdot n^{20} \Rightarrow O(n^{20})$

$c_2 = 2^{10}, n_{02} = 1$

(ii) $(n^2+1)^{10} \ge (n^2)^{10}$

$\ge n^{20}$

$\Rightarrow \Omega(n^{20})$

$c_1 = 1, n_{01} = 1$

from (1), (2)

$\therefore (n^2+1)$ is $\Theta(n^{20})$

$c_1 = 1, c_2 = 2^{10}, n_0 = 1$

(1)

ASS 1 - female Feb-2022

Q1

(a) it computes the sum of squares of first n integers

$$S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2 = \sum_{i=1}^{n} i^2$$

(b) The Basic operation

وهي العملية الأساسية التي يتم تكرارها مع كل لفة

is Multiplication

or

Multiplication and addition

(c) عدد مرات تكرار او تنفيذ العملية الأساسية

هي عدد مرات تكرار اللوب $= n$

(d) $\Theta(n)$

(e) سبق ان حسبنا قاعدة لمجموع تربيعات الأعداد

$$S(n) = 1^2 + 2^2 + \dots + n^2 = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

الحل

Data: A nonnegative integer n

$s \leftarrow n * (n+1) * (2*n+1)/6$

return s

QB| Consider the following algorithm.

Algorithm $Secret(A[0..n-1])$
//Input: An array $A[0..n-1]$ of n real numbers
$minval \leftarrow A[0]$; $maxval \leftarrow A[0]$
for $i \leftarrow 1$ **to** $n-1$ **do**
 if $A[i] < minval$
  $minval \leftarrow A[i]$
 if $A[i] > maxval$ Q3
  $maxval \leftarrow A[i]$
return $maxval - minval$

solution

a. Computes the range, i.e., the difference between the array's largest and smallest elements.

b. An element comparison.

c. $C(n) = \sum_{i=1}^{n-1} 2 = 2(n-1)$.

d. $\Theta(n)$.

e. An obvious improvement for some inputs (but not for the worst case) is to replace the two if-statements by the following one:

if $A[i] < minval$ $minval \leftarrow A[i]$
else if $A[i] > maxval$ $maxval \leftarrow A[i]$.

Another improvement, both more subtle and substantial, is based on the observation that it is more efficient to update the minimum and maximum values seen so far not for each element but for a pair of two consecutive elements. If two such elements are compared with each other first, the updates will require only two more comparisons for the total of three comparisons per pair. Note that the same improvement can be obtained by a divide-and-conquer algorithm.

HW1 - Q5 - b

(i) $\sqrt{10n^2+7n+3}$

$\leq \sqrt{10n^2+7n^2+3n^2}$

$\leq \sqrt{20n^2}$

$\leq \sqrt{20} \cdot \sqrt{n^2}$

$\leq \sqrt{20} \cdot n \Rightarrow O(n)$

$c_2 = \sqrt{20}, n_{02} = 1$

(ii) $\sqrt{10n^2+7n+3}$

$\geq \sqrt{10n^2}$

$\geq \sqrt{10} \cdot \sqrt{n^2}$

$\geq \sqrt{10} \cdot n \Rightarrow \Omega(n)$

$c_1 = \sqrt{10}, n_{01} = 1$

from 1, 2

$\therefore \sqrt{10n^2+7n+3}$ is $\Theta(n)$

$c_1 = \sqrt{10}, c_2 = \sqrt{20}, n_0 = 1$